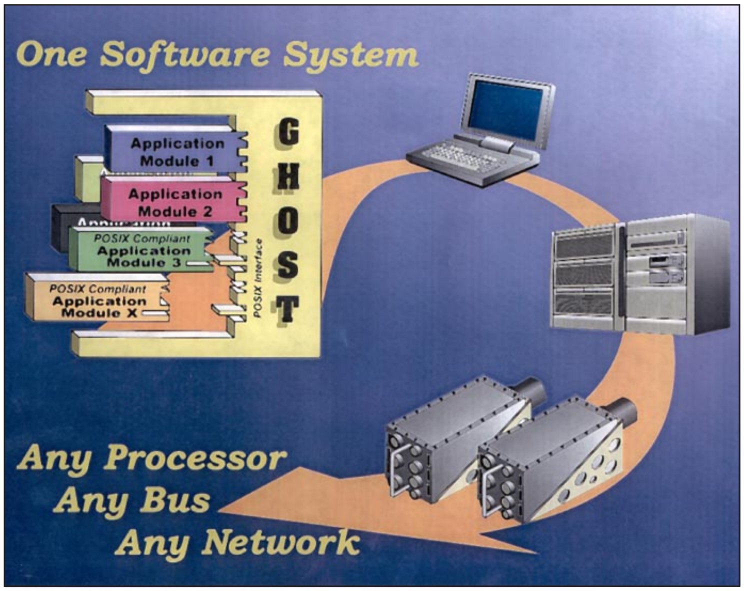
Air Force Operating System Software Converted to Commercial Product



Payoff

By leveraging the capabilities and benefits of existing Air Force operational flight program software to meet the needs of the commercial world, American Computational Technical Services can now provide commercial/government users a versatile and flexible real-time application environment. Their Generic Host-independent Operating System runs on a variety of computer systems, permitting the user to develop applications in familiar environments and reducing development time. Code portability further reduces or eliminates the amount of debugging needed to move code to new platforms.

Accomplishment

A Cooperative Research and Development Agreement between the Information Directorate's Collaborative Simulation Technology and Applications Branch (previously the Avionics Directorate's Avionics Simulation Technology Branch) and American Computational Technical Services of Huber Heights OH, resulted in the transfer of an Air Force developed, real-time reconfigurable multiprocessor operating system to the commercial sector. By converting Ada Avionics Real-Time Software designed for embedded avionics flight processors to operate generically on a variety of commercial computing platforms, American Computational now provides a commercial version.

Background

The new commercial product is based on the Air Force's Ada Avionics Real-time Software and is called Generic Host-independent Operating System (GHOST). It offers a programming and execution environment for the development of new application software and for the reuse of Ada and legacy code. GHOST runs on a variety of computers including embedded flight processors, workstations and personal computers, and it also incorporates the computer industry standard Portable Operating System Interface. Applications written using GHOST's constructs and services are easily ported to new hosts minimizing the interaction between dissimilar computers. It takes advantage of available (hardware and programming) resources and enhances software reuse, reducing the software life-cycle costs. Because of its transparency, users can construct networks of similar computers. Code can be developed and tested in a user friendly environment, significantly reducing the amount of code debugging needed in the more restrictive environment.